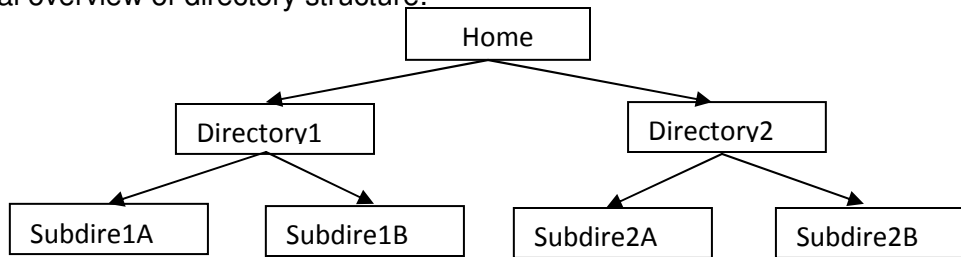


Linux directory structure and commands

Graphical overview of directory structure:



Important considerations:

Linux commands and filenames are case sensitive. Hence 'ls' is not the same as 'LS' and protein.mol2 is not the same as Protein.mol2.

Do NOT use spaces in directory names or filenames. You can use underscores (_), dashes (-), and dots (.). You can have an unlimited number of directories and subdirectories and levels.

When you first log on to a remote machine, you will be in your home directory. You can see this by typing the command for print working directory (the one you are in; after typing commands press the return key):

`$pwd`

This should return something like: `/home/chem204/kavanagh`

Here are a few basic commands that are often used when using a terminal window (also known as a shell):

- `pwd` - show present working directory
- `ls` - list files in present directory.
- `ls /home/chem204` - list files in the directory `/home/chem204`. Can list files in any directory that you are allowed access to by giving its path name.
- `mkdir dir` - create a directory
- `cd dir` - change directory to dir. If you type just `cd`, it will take you to your home directory
- `cd ..` - go up one directory.
- `history` - will print out a history of all the commands you have typed
- `clear` - clear the screen
- `cp file1 file2` - copy file1 to file2, can easily copy files from one directory to another.
- `mv file1 file2` - moves file1 to a file named file2. (used to rename files)
- `less file1` - view the contents of file1, the PageUp/PageDn keys can be used to scroll through the file.
- `rm file1` - remove (delete) a file – *careful! It won't ask for verification and there is no 'undo'.*
- `rmdir dir` - delete directory dir
- `cat file1` - display the contents of a file on screen
- `↑` (or `↓`) - using the up arrow will show the commands that you previously typed. Very useful if you want to rerun a command.

Many commands will need some type of argument, such as which directory to change to or where to copy or move a file to/from. These are shown in italics above. There needs to be a space between the command and each argument as shown above or you will get an error message.

If you want information about how to use a command or what options there are, you can see the manual or help pages. If you type `help`, it will display the output to the screen and exit. If you type `man`, you can scroll through the page by using the PageUp/PageDn (or Ctrl-u and Ctrl-d) keys. To close the man page, type q (for quit).

```
$ls --help
```

```
$man ls
```

Organizing your home directory

If you keep all your work in your home directory, it will quickly become full of files and difficult to find the ones you are interested in.

What you want to do is organize the files by the library you are screening. The easiest way to do this is to make a new directory for each library.

As an example, make a directory for CB-kin_UT and move most of files that you produced last week into it. For the files that you will use again for docking of different libraries, copy them into the new directory, but leave the originals in your home directory.

```
$mkdir CB-kin_UT
```

```
$mv bestrank_CB-kin-UT_run1.lst CB-kin_UT_run1.sdf bestrank_CB-kin-UT_run2.lst CB-kin_UT_run2.sdf *err *log *out CB-kin_UT/.
```

```
$cp *mol2 *conf CB-kin_UT/.
```

The '*' is a wildcard that tells the operating system to copy or move all files with the given extension. The '.' after the `CB-kin_UT/` tells Linux to keep the same name for the files it is copying or moving.

The '.' character is also a shortcut that means the current directory, some of you saw this in your .conf file. If you type '..' that refers to one directory up from the current directory.

For future work, it is easier to make a directory that is named according to the library that you will be screening and submit the job from that directory so that the files will be produced in the new directory.

```
$mkdir CB_MW-set
```

 (Make a directory for the library you are screening, it isn't necessary to make one for each of the numbered files in the library)

```
$cp *mol2 *conf gold.hosts CB_MW-set/.
```

 (Copy the files that you need to run GOLD into the new directory)

```
$cd CB_MW-set
```

 (change directories so that you are in the new directory where you will run GOLD)

Use SFTP to edit your configuration file paying particular attention to the following lines:

```
autoscale = 0.5                (autoscale= 0.5 for run1, for run2 autoscale = 1.0)
ligand_data_file /home/chem204/MayBridge_1.sdf 10 (the library you want to screen or the output
from run1 if you are doing run2)
bestranking_list_filename = MayBridge_1_run1.lst (match the name of the file to the library and run)
concatenated_output = MayBridge_1_run1.sdf (match the name of the file to the library and run)
clean_up_option save_best_ligands 500          (500 for run1, 10 for run2)
```

Once the .conf file has been edited correctly you can start GOLD. **Make sure that you change the bestranking_list_filename and concatenated_output filenames before running each new job or the program will overwrite the older files and the work will be lost.**

```
$goldremoteP 6 #####.conf &
```

 (start your GOLD job)